

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-222000

(43)Date of publication of application : 18.08.1995

(51)Int.Cl.

H04N 1/44

G09C 1/00

H04N 1/00

(21)Application number : 06-308029

(71)Applicant : AT & T CORP

(22)Date of filing : 13.12.1994

(72)Inventor :
BRASSIL JOHN T
LOW STEVEN H
MAXEMCHUK NICHOLAS F
O'GORMAN LAWRENCE P

(30)Priority

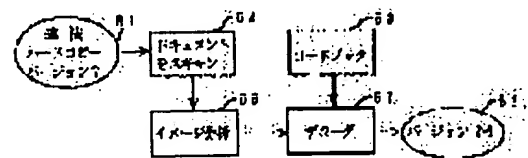
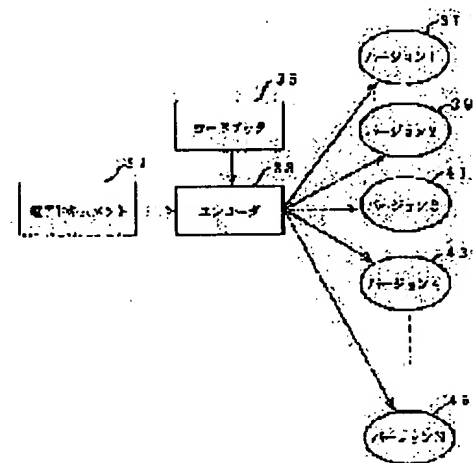
Priority number : 93 170619 Priority date : 20.12.1993 Priority country : US

(54) DOCUMENT COPY PREVENTING METHOD

(57)Abstract:

PURPOSE: To provide a method for preventing the illegal copying and distribution of a document.

CONSTITUTION: A process for using a computer system for electronically issuing the plural copies of the document distributed for plural subscribers and a process for executing a program in a computer and executing an identification code function are contained. The processes are for encoding the plural copies with peculiar identification codes. The identification codes are based on the peculiar arrangement of the electronically generated materials of the copies and code books which associate the special subscribers with the identification codes are generated. A decoding method is contained in the ability of encoding. The peculiar arrangement of the material which is electronically generated is based on line shift encoding, word shift encoding and feature reinforcement encoding (or the combination of them) and it is brought by a bit map change or a document format file change.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-222000

(43) 公開日 平成7年(1995)8月18日

(51) Int.Cl. ⁴	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N 1/44				
G 0 9 C 1/00		9364-5L		
H 0 4 N 1/00	1 0 2 B			

審査請求 未請求 請求項の数27 O L (全 18 頁)

(21) 出願番号 特願平6-308029

(22) 出願日 平成6年(1994)12月13日

(31) 優先権主張番号 1 7 0 6 1 9

(32) 優先日 1993年12月20日

(33) 優先権主張国 米国 (U S)

(71) 出願人 390035493

エイ・ティ・アンド・ティ・コーポレーション

AT&T CORP.

アメリカ合衆国 10013-2412 ニューヨ

ーク ニューヨーク アヴェニュー オブ
ジ アメリカズ 32

(72) 発明者 ジョン テー, ブラジル

アメリカ合衆国, 07922 ニュージャージー,
パークレイ ハイツ, エヴァーグリーン
ドライブ 47

(74) 代理人 弁理士 岡部 正夫 (外2名)

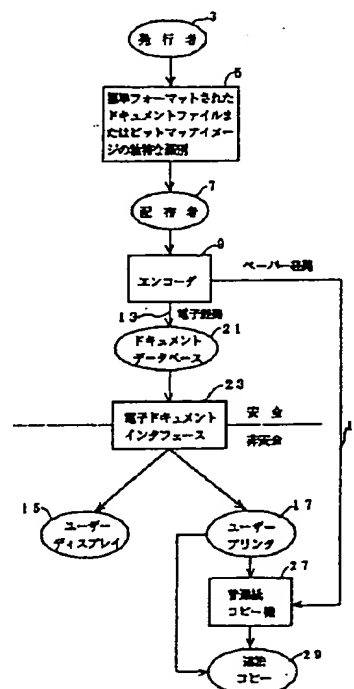
最終頁に続く

(54) 【発明の名称】 ドキュメントコピー防止方法

(57) 【要約】 (修正有)

【目的】 ドキュメントの違法コピーおよび配布を防止する方法を提供する。

【構成】 複数の加入者に配布するためのドキュメントの複数のコピーを電子的に発行するためコンピュータシステムを利用する工程と、コンピュータ内でプログラムを実行して識別コード機能を行う工程とを含む。これらの工程は、それぞれ独特な識別コードで複数のコピーをコード化するためのものであり、その識別コードは、コピーの電子的に生成された材料の独特の配置に基づくものであり、特別な加入者へそのような識別コードを相互に関連づけるコードブックを作成する。解読方法はコード化の能力に含まれている。電子的に生成された材料の独特の配置は、ラインシフトコード化、ワードシフトコード化、もしくは特徴強調コード化（または、これらの組み合わせ）に基づくものであり、ビットマップ変更もしくはドキュメントフォーマットファイル変更によりもたらされている。



【特許請求の範囲】

【請求項1】 電子発行されたドキュメントの違法コピーを防止する方法において、

(a) 複数の加入者に配布するため電子的に生成された材料を有するドキュメントの複数のコピーを電子的に発行するためにコンピュータシステムを使用し、

(b) 前記コンピュータシステム内でプログラムを行い、(i) それぞれ、別々の特有の識別コードを有する複数のコピーをコード化する工程と、前記識別コードは、各コピーに電子的に生成された材料の独特の配置に基づくものであり、(ii) 特定の加入者にそのような識別コードをそれぞれ相互に関連づけるためのコードブックを作成する工程とを実行することからなることを特徴とする防止方法。

【請求項2】 前記プリントした材料はテキストの材料のラインであり、電子的に生成された材料の前記特有の配置は、ラインシフトコード化に基づくものであることを特徴とする請求項1記載の方法。

【請求項3】 前記ラインシフトコード化は、ドキュメントの各コピーを独特にコード化するためにドキュメントに含まれたその他のラインと比較して少なくとも1つのラインの位置をシフトするようにドキュメントフォーマットファイルを変更することによって成し得ることを特徴とする請求項2記載の方法。

【請求項4】 前記ラインシフトコード化は、ドキュメントの各コピーを独特にコード化するためにドキュメントに含まれたその他のラインと比較して少なくとも1つのラインの位置をシフトするようにドキュメントのビットマップイメージを変更することによって成し得ることを特徴とする請求項2記載の方法。

【請求項5】 前記電子的に生成する材料は、特定の順序で配置されたワードを含み、材料の前記独特の配置は、ワードシフトコード化に基づいていることを特徴とする請求項1記載の方法。

【請求項6】 前記ワードシフトコード化は、ドキュメントの各コピーを独特にコード化するためにドキュメントに含まれたその他のワードと比較して少なくとも1つのワードの位置をシフトするようにドキュメントフォーマットファイルを変更することによって成し得ることを特徴とする請求項5記載の方法。

【請求項7】 前記ワードシフトコード化は、ドキュメントの各コピーを独特にコード化するためにドキュメントに含まれたその他のワードと比較して少なくとも1つのワードの位置をシフトするようにビットマップイメージを変更することによって成し得ることを特徴とする請求項5記載の方法。

【請求項8】 前記電子的に生成する材料は、標準化したプリントの特徴を含み、材料の前記独特の配置は、特徴変更コード化に基づいていることを特徴とする請求項1記載の方法。

【請求項9】 前記特徴変更コード化は、前記標準化したプリントの特徴と比較して少なくとも1つのプリントの特徴を変更するようにビットマップイメージを変更することによって成し得ることを特徴とする請求項8記載の方法。

【請求項10】 前記プログラムを実行するため、(i) (ii) 標準のドキュメントとして、前記ドキュメントの第1のコピーを作成する工程と、(iv) 前記標準ドキュメントとは異なるものとする少なくとも一つの変更をそれぞれ有する複数のその後のコピーを作成し、前記コピーが少なくとも前記ひとつの変更に基づいてそれぞれ独特の識別コードを有するように互いに異なるものとする工程と、(v) 前記標準ドキュメントと比較してそのような各コピーの一連の同じおよび異なる点を認識するため前記標準ドキュメントと各その後のコピーとを比較する工程と、(vi) 独特なバイナリの識別コードに対する前記一連の同じおよび異なる点をコンバートする工程とを実行することを含む請求項1記載の方法。

【請求項11】 前記コンピュータシステムは、それに接続されたスキャナ装置を有し、前記プログラムは、

(1) 前記コンピュータシステムへイメージを供給するためドキュメントのコピーをスキャンする工程と、

(2) その独特な識別コードを決定するためイメージを分析し、解読する工程と、

(3) 識別コードが相互に関連する特別な加入者を決定するためコードブックと結果として生じた識別コードとを比較する工程と、を実行することを含むことを特徴とする請求項1記載の方法。

【請求項12】 前記プログラムは、

(1) 予め電子的に発行されたドキュメントのコピーのビットマップイメージを受け取る工程と、

(2) その独特な識別コードを決定するためビットマップイメージを分析する工程と、

(3) 識別コードが相互に関連する特別な加入者を決定するためコードブックと結果として生じた識別コードとを比較する工程とを実行することを含むことを特徴とする請求項1記載の方法。

【請求項13】 前記プログラムは、

(1) 予め電子的に発行されたドキュメントのコピーのドキュメントフォーマットファイルを受け取る工程と、

(2) その独特な識別コードを決定するためドキュメントフォーマットファイルを分析する工程と、

(3) 識別コードが相互に関連する特別な加入者を決定するためコードブックと結果として生じた識別コードとを比較する工程とを実行することを含むことを特徴とする請求項1記載の方法。

【請求項14】 前記ドキュメントのコピーは、ラインシフト変更によりコード化されたドキュメントからのものであることを特徴とする請求項11記載の方法。

【請求項15】 前記プログラムは、ノイズ除去の工程

も有することを特徴とする請求項14記載の方法。

【請求項16】 前記分析および解読する工程は、ベースライン誤差決定に基づいていることを特徴とする請求項14記載の方法。

【請求項17】 前記分析および解読する工程は、中心誤差決定に基づいていることを特徴とする請求項14記載の方法。

【請求項18】 前記ドキュメントのコピーは、ラインシフト変更によりコード化されたドキュメントからのものであることを特徴とする請求項13記載の方法。

【請求項19】 前記プログラムは、ノイズ除去の工程も有することを特徴とする請求項18記載の方法。

【請求項20】 前記分析および解読する工程は、ベースライン誤差決定に基づいていることを特徴とする請求項18記載の方法。

【請求項21】 前記分析および解読する工程は、中心誤差決定に基づいていることを特徴とする請求項18記載の方法。

【請求項22】 前記ドキュメントのコピーは、ワードシフト変更によりコード化されたドキュメントからのものであることを特徴とする請求項11記載の方法。

【請求項23】 前記ドキュメントのコピーは、ワードシフト変更によりコード化されたドキュメントからのものであることを特徴とする請求項12記載の方法。

【請求項24】 前記ドキュメントのコピーは、ワードシフト変更によりコード化されたドキュメントからのものであることを特徴とする請求項13記載の方法。

【請求項25】 前記ドキュメントのコピーは、特徴強調変更によりコード化されたドキュメントからのものであることを特徴とする請求項11記載の方法。

【請求項26】 前記ドキュメントのコピーは、特徴強調変更によりコード化されたドキュメントからのものであることを特徴とする請求項12記載の方法。

【請求項27】 前記ドキュメントのコピーは、特徴強調変更によりコード化されたドキュメントからのものであることを特徴とする請求項13記載の方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、各加入者特有の独特な識別コードを作成することにより、電子発行されたドキュメントの違法コピーを防止する方法に関するものである。発行されたドキュメントの各コピーは、たとえばプリント材料またはディスプレイ材料などの電子的に生成された材料の独特の配置を有し、これは、訓練されていない人間の目には即座に見分けることのできないものである。それらの独特な識別コードによって、違法コピーを防ぎ、また発行者および著作権者が違法コピーを分析できるようにし、その元の加入者を判断することができる。

【0002】

【従来の技術】 コピー機による複写の質が、オリジナルと匹敵しうようになり、コピーの費用は、1ページにつき2、3ペニーに削減され、また1ページをコピーするのに要する時間も1秒もしくはそれ以下に削減された時、コピー機は、発行者を脅かす存在となった。この種の問題は、電子領域においてより大きいものとなっている。複写の質がオリジナルと一致し、コピーを作成するのにほとんど費用がかからず、たったひとつのキーストロークで、何百ページというコピーが、数秒でできてしまう。加えて、送信者は、ほとんど何の努力もせずに、電子ドキュメントを電子メールやネットワークの新たなサービスによって、多人数のグループに配布することができる。

【0003】

【発明が解決しようとする課題】 ところで、簡単に且つ費用がかからずコピーでき、電子ドキュメントを配布できることは、電子発行が従来の発行に実際にとってかわるようになる前に解決されなければならない主要な技術的課題と考えられている。しかしながら、個人が、個人の占有にあるデータファイルを複写されないようにすることは、不可能ではないとしても、非常に困難なことである。そこで、本発明は、一般のデータファイルの複写を防止する試みに代わって、ドキュメントのテキスト部分のビットマップにおいてオリジナルの所有者を確認できるようにすることで電子発行をより充実したものとするを目的としている。現在の著作権法に関して、本発明は、他で起きるであろう数多くのコピーおよび配布を防止するに十分なものでなければならない。本発明の方法による興味深い結果は、また、発行者もしくは著作権者が、複写されたコピーが見つかったときにそのオリジナルが誰に属するのかを判断することができることである。本発明の技術的課題および一般的な目的は、ドキュメントの違法コピーおよび配布を防止する方法を提供することにある。

【0004】

【課題を解決するための手段】 本発明は、電子的に発行されたドキュメントの違法コピーを防止する方法に関するものである。本発明の方法は、複数の加入者に配布するための電子的に生成された材料を有するドキュメントの複数のコピーを電子的に発行するためにコンピュータシステムを利用する工程と、コンピュータ内でプログラムを実行して識別コード機能を行う工程とを含む。これらの工程は、それぞれ独特な識別コードで複数のコピーをコード化するためのものであり、その識別コードは、そのようなコピーの電子的に生成された材料の独特の配置に基づくものであり、特別な加入者に対してそのような識別コードを相互に関連づけるコードブックを作成する。いくつかの実施例においては、解読方法はコード化の能力に含まれている。電子的に生成された材料の独特の配置は、ラインシフトコード化、ワードシフトコード

化、もしくは特徴強調コード化（または、これらの組み合わせ）に基づくものであり、ビットマップ変更もしくはドキュメントフォーマットファイル変更によりもたらされている。

【0005】

【実施例】以下に本発明の実施例を説明する。本発明の方法においては、ドキュメント特徴づけは、配布されるドキュメントの各コピー内に独特な識別コードを埋め込み、特別な加入者（受信者）へ識別コードを相互に関係づけるコードブックが保持される。したがって、復元されたドキュメント（時として、記述ドキュメントのコピー）を検査することで、オリジナルのドキュメント受信者の身元が判る。ドキュメント特徴づけは、テキストのフォーマットすなわち、行（以下、「ライン」とする）、語（以下、「ワード」とする）、または文字のまとまりの変更、あるいはテキスト構成要素のある文字の変更（たとえば、個々の文字の変更）のどちらかにより行われる。本発明の方法におけるドキュメントを特徴づけるのに用いるこれらの変更により、発行者は、（1）

保安（トレース能力）目的のため、識別されるコードワードをはめ込み、（2） できる限り見た目に変化のない特徴を変更することができる。

【0006】あるタイプの特徴づけは、プリントやスキャン、普通紙コピーなどによりドキュメントに生じるノイズの存在において発見される。本発明で用いられる「コード化された」ドキュメントは、次のものを含むいくつかの可能な方法において保安性を提供することができる。

（1） ドキュメントは、各位置、加入者、受信者、または使用者（以下、「加入者」とする）のために特別にコード化される。これにより、意図する加入者以外へのコード化されたドキュメントのいかなる配布も意図する加入者へトレースすることができる。

（2） ドキュメントコードは、ユーザーインターフェイス（たとえば、特別な加入者のコンピュータワークステーション）の特別なインストールへ正當に適合するようにドキュメントを特徴づける。このインターフェイスへ適合しないドキュメントをディスプレイしようとする、そのインターフェイスは、ドキュメントのディスプレイを拒絶するなどの方法で形成される。

【0007】1. 0 アプリケーションの概略

本発明によるドキュメント作成、配布およびユーザー相互作用の概略は、図1に図示されている。これは、ドキュメントが発行者3からユーザーに流れることのできる3つの経路が示されている。第1の経路は、通常のペーパーコピー配布チャネル11（すなわちユーザーが発行者から紙の刊行物を受けること）である。第2の経路および第3の経路は、ユーザーのディスプレイ15のためドキュメントデータベース21および電子ドキュメントインターフェイス23を介する、もしくは、プリント

されたドキュメントを作成するためユーザーのプリンター17を介する電子配布経路13である。ペーパーコピー配布チャネル11から、あるいはユーザーのプリンターからであろうと、普通紙コピー機27は、たとえば、違法のペーパーコピー29を作り出すために使用される。もちろん、本発明の範囲を逸脱しない限り、図1のフローチャートには、さまざまな変更をなし得ることができる。たとえば、違法のユーザーは、スキャナを用いて合法バージョンをスキャンし、その後、電子的に違法のコピーを再生できる。本発明の方法は、これらのあるいは類似したタイプの配布経路、たとえば、電子的に発行されたものやファックスや無線通信のコンピュータなどにより配布されたもののどれかに適応できるドキュメントに及ぶ。ドキュメントのコード化は、エンコーダ9により示されたようにドキュメント配布の前に行われる。

【0008】ドキュメントは、電子フォームの間にコード化される（図2）。ドキュメントをコード化する技法は、イメージもしくはフォーマットされたドキュメントファイルの2つのフォームのどちらかで行われる。イメージ表示は、ピクセルの配列としてドキュメントの各ページ（または、サブページ）を記す。イメージは、白黒（ビットマップともいう）、グレースケール、もしくはカラーである。このテキストの残りにおいて、イメージ表示は、イメージカラーの中味にかかわらず、「ビットマップ」として、単純に参照される。フォーマットされたドキュメントファイルの表示は、ポストスクリプト、トロフ、SGMLなどの標準フォーマット記載言語などを用いて、ドキュメントの中味を記載するコンピュータファイルである。

【0009】一般的な応用では、ビットマップは、フォーマットされたドキュメントファイルから発生する。ドキュメントを特徴づけるため本発明で用いられたコード化の技法は、エンコーダに提供されたオリジナルのフォーマット、および加入者が見るフォーマットに、ある程度依るものである。加入者が、あるドキュメントを見る（たとえば、ワークステーションのモニターで1ページ表示する）と、その加入者は、そのドキュメントを捕捉でき違法に配布できると考える。したがって、コードは、この加入者の段階の前にコード化されなければならない。図2に示されるように、電子ドキュメント31は、コードブック35で設定された変更の前もって選択された設定にしたがってエンコーダ33でコード化される。

（コードブックが、コード化よりも前にあることは必然的ではない。特定の実施例では、コードブックは、特定の加入者にそれらを相互に関連づけるために、使用される識別コードのロギングから作り出される。）コード化されたドキュメントは、バージョン1（37）、バージョン2（39）、バージョン3（41）、バージョン4（43）……バージョンN（45）までのように、

それぞれ固有に作り出される。

【0010】本発明の方法では、さまざまなコード化の技法が用いられる。そして、それらは、文字、グラフィック、アルファベット、数字や他の固有の識別名を用いる必要なく、また、これによってコードに違法のコピーを登録することなく、ライン、ワードまたは文字の特徴の変更（もしくは、これらの組み合わせ）に関連している。したがって、すべての方法に共通することは、すでに存在する特徴の特別な態様を変更することによって、ドキュメントにおいてコードワードをコード化することである。たとえば、コードワード1101（バイナリ）について考えてみる。このコードを右から左へ最小の意味のあるビットから読んでいくと、第1のドキュメント特徴は、ビット1に変更され、第2の特徴は、ビット0に変更されず、そして次の2つの特徴は、2つの1ビットに変更される。それぞれの特別なコード化方法を区別するのは、特徴のタイプである。

【0011】(1) ラインシフトコード化方法 — ドキュメントを固有にコード化するためテキストのラインの位置をシフトすることによって、ドキュメントフォーマットファイルを変更する方法である。このコードは、フォーマットファイルまたはビットマップから解読される。ラインは、たとえば、水平にもしくは垂直にぶれる。この方法は、ノイズによって品質の低下したイメージの中にある場合でも、コード検出のためのこれらの方法の中でも、高い信頼性を提供することができる。

(2) 特徴強調コード化方法 — ドキュメントを固有にコード化するため、あるテキスト要素の特徴を修正することによって、ドキュメントのビットマップイメージを変更する方法である。そのような修正の一例は、文字のアセンダの長さを延ばすことである。他では、文字幅を狭めること、また、文字セクションを削除するかもしくは短くすることである。このタイプのコードは、ビットマップイメージからコード化され、また解読される。

(3) ワードシフトコード化方法 — ドキュメントを固有にコード化するため、テキスト内のワードの位置をシフトすることによって、ドキュメントフォーマットファイルやイメージビットマップを変更する方法である。このコード化は、フォーマットファイルもしくはビットマップから解読されるであろう。ドキュメントフォーマットファイルの変更を用いる好ましい実施例におけるこの方法は、方法(1)に使用されるものと似ている。この方法は、一般的に、方法(1)よりも、(視角的に)より分らないドキュメントの変更を提供する。しかしながら、ノイズの多いイメージからの解読は、より困難となる。

上述の3つのコード化技法を以下に詳細に論じる。

【0012】2.1 テキストラインコード化方法

これは、フォーマットされたドキュメントファイルに適

応されるコード化方法である。次の記述において、フォーマットされたドキュメントファイルは、ポストスクリプト(今日、最も一般的なページ・デスクリプション・ランゲージフォーマットである)と一体となった、アドビ(Adobe)システムにおけるものであると仮定する。しかしながら、本発明は、プログラムをフォーマットする他のドキュメントファイルにも適用できる。ポストスクリプトは、ドキュメントの中味を一度に1ページ記述する。単に、置く、これは、テキストライン(もしくは、フレーズ、ワードまたは文字などのテキストラインの一部)の中味を特定し、ディスプレイされるテキストの位置を確認する。テキストの位置は、仮想ページ上の位置を表わすX-Y軸で特徴づけられる。ポストスクリプトを作り出すソフトウェアによって用いられる解像度によって、テキストの位置は1/720インチ(プリンターの「ポイント」の1/10)ほどの小ささに修正される。今日最も一般的に使用されるレーザープリンタは、さらにやや小さな解像度(たとえば、1/3000インチ)を有する。

【0013】本発明の方法のひとつの実施例では、配布の前に、オリジナルのポストスクリプトドキュメントおよびコードワードはエンコーダに供給する。エンコーダは、コードワードを読み、動かすべきラインを探す。動かすべきラインを見つけたと、エンコーダは、ラインスペース調整に組み込むように、オリジナルの(間隔をあけていない)ポストスクリプトファイルを修正する。これは、スペースをあけるべきラインのY軸を増加したり減少したりすることによりなされる。エンコーダの出力は、加入者に電子的またはペーパーフォームで配布できるよう用意された「コード化された」ポストスクリプトドキュメントである。

【0014】図3は、発行者が、どのようにして、ドキュメントの復元されたペーパーコピーの分析によって特徴づけられたドキュメントのオリジナルの受信者(加入者)を確認するかを図示している。これは、疑わしきハードコピー51があり、コピー51は、スキャナ53によってスキャンされ、コンピュータ55によって分析され、解読プログラム57で解読され、コードブック59と合わされて、元のあるいは加入者のバージョン61を決定する。たとえば、「デコーダ」は、ライン間隔を分析し、オリジナルの加入者を固有に認識する対応するコードワードを抽出する。

【0015】ドキュメントの違法コピーのページ(1ページもしくは複数ページ)は、ページのビットマップイメージを作成するため、電子的にスキャンされたものでもよい。ビットマップイメージは、好ましくは、あるタイプの外部の特徴づけ(すなわち、ハードコピーのプリント、普通紙コピー、電子スキャン、しみなどにより生じるノイズ)を取り除くためノイズ削除を受ける。ビットマップイメージは、テキストラインがサイドのページ

の端と垂直であることを確実にするために回転される。ページの「プロフィール」は、イメージ内の水平なスキャンラインごとのビット数として見出しされる。我々の経験においてであるが、そのようなスキャンラインの数は、テキストラインにつき40あたりで変動する。隣り合うテキストラインのプロフィール間の各間隔が測定される。これは、隣り合うラインのプロフィールのベースライン間の間隔が測定される、あるいは隣り合うラインのプロフィールの中心（すなわち、かたまりの中心）間の間隔が測定されるかの2つの方法のうちのひとつによって行われる。このインターライン間隔は、分析されて、間隔が加えられたか減じられたかが判断される。このプロセスは、各ラインごと繰返され、ドキュメントのコードワードが判断されて、オリジナルの加入者が固有に決定される。

【0016】本発明の他の方法と比較してこの方法の利点は、次の通りである。オリジナルなしでコードが解読される。このため、解読が非常に簡単である。これは、最もノイズに抵抗できる技法であると思われる。しかしながら、この方法は、ここで記述されるコード化の技法のうちで最も視覚的にわかってしまう方法であると考えられる。

【0017】図4に、ポストスクリプトファイルの単純なライン間隔エンコード疑似コードを図示した。図5に、復元されたドキュメントのページのライン間隔プロフィールのグラフを示した。各テキストラインのベースラインのスキャンラインは、「+」で示される。各テキストラインの中心のスキャンラインは、ドットで示される。ライン間隔でページを解読するには、隣り合うテキストラインの中心もしくはベースライン間の間隔を測定し、間隔が増加されているか、減少されているか、もしくは標準と同じままかどうかを判断する。

【0018】2. 2 特徴強調コード化方法

この方法は、ドキュメントのビットマップイメージへ直接適応される本発明のコード化方法である。ビットマップイメージは、選ばれた特徴が検討される。そしてそれ

DECODING:

```
read in list of codeword bits and corresponding line
numbers
      and space locations in text-lines
      where coding has been performed on original
      image
set codedImage = 1
for each text-line in coded formatted file and original
formatted file
{
  if codeword bit for a text-line is 1
  then
    if coded spaces in coded image are not
    different from corresponding
    spaces in original image
    then codedImage = 0, break
}
if codedImage = 1, then image matches code
else if codedImage = 0, then image does not match code
```

【数2】

らの特徴が変更されるか、変更されないかは、コードワードによるものである。これらの変更は、広げたり、狭めたり、斜めにしたり、個々の文字の特徴を減じたり、付加したりすることにより行われる。たとえば、文字の上方の垂直なエンドライン、すなわち、b、d、hなどの文字の上部が延ばされる。これらのエンドラインは、その長さを1ピクセル（もしくはそれ以上）延ばすことにより変更し、他は何の変更もしない。

【0019】このコード化は、ビットマップイメージに適用され、またプリンタイメージでも検出される。以下に示唆するものよりもさらに強調したコード化、または、コード化における冗長度により、プリントされたおよび写真複写されたドキュメントのスキャンされたイメージにおいても検出される。本発明のこの方法の利点は、次の通りである。非常に数多くのコード化の可能性がある（おそらく、ワードシフトコード化の10倍、また、ラインシフトコード化の20倍）。このコード化は、ドキュメントのビットマップに行われる。そして、フォーマットされたドキュメントファイルを変更する必要がない。イメージをコード化する方法のうちで最も視覚的にわかりやすい方法のひとつである。

【0020】欠点は、次の通りである。この方法は、主として、イメージコード化の技法であり、通常フォーマットファイルには適用できない（他の技法は、両方にさらに容易に適用できる）。この方法は、ノイズに耐えられないような大きさ（長さ）でコード化する時、その可視度により検知するものであるため、写真複写された、またはそれ以外でもノイズの多いドキュメントにはあまり適用できないものと考えられる。このコードは、オリジナルなしでは検出を行うことができない。

【0021】コード化や解読のための疑似コードは、図6に関して、次の通りである。図6には、5×5のピクセル配列の通常のコード化63、65および67の3つの例と、強調コード化73、75および77が示されている。

【数1】

//

12

CODING:

```

mask off the least significant codeword bit and right-
shift the codeword
for each text line in the format file
{
  if the code bit is 1
  {
    find the longest space between words
    find the shortest space between words
    shorten longest space and lengthen shorter space
    by a chosen amount (must be <= longest space-
    shortest space)
    store text-line number, altered space
    positions, and codebit
    if codeword = 0, break
    else mask next codeword bit and
    right-shift
  }
}

```

【0022】

(参考)

コード化:

選ばれた命令 (たとえば、ラスタスキャン命令) におけるイメージの各ピクセルごとに最小の意味のあるコードワードをマスクし、そのコードワードを右にシフトする

{

図6に示される選ばれた特徴のひとつと適合する場合、

この中心ピクセルの周りのピクセルの $k \times k$ (たとえば、 5×5) 隣を検査
 $k \times k$ マスク内のピクセルのパターンが、

{

コードワードビットが1である場合、図6に示された特徴を変更する

もしくは、コードワードビットが0である場合、コードワードの中心ピクセル

と1もしくは0値との記憶 (X, Y) 位置である特徴を残す

コードワードが0である場合、ブレイクする

もしくは、次のコードワードビットをマスクし、

右にシフトする

}

}

【0023】

解説:

コードワードビットのリストおよび対応する中心ピクセルの位置を読み取る

ここでは、コード化は、オリジナルイメージ上で行われる

コード化された特徴の各 (X, Y) 位置ごとに

コードイメージ=1に設定

{

$k \times k$ 範囲が、パターンと一致し、コードワードビットが0である場合、

または、ピクセルは、パターンが変更されておらず、かつ

コードワードビットが1の場合、

中心ピクセルの周りのピクセルの $k \times k$ 隣を検査

コードイメージ=0となる、ブレイクする

}

コードイメージ=1の場合、イメージは、コードと一致する

コードイメージ=0の場合、イメージは、コードと一致しない

【0024】2. 3 ワードシフトコード化方法

るドキュメントに適応しえるコード化の方法である。こ

この方法は、隣り合うワード間の変動しえる間隔を有す 50 のコード化は、フォーマットファイルに最も容易に適応

13

される。各テキストラインごとに、ワード間のもっとも大きい間隔と最も小さい間隔とがを見つけ出される。ラインをコード化するために、最も大きな間隔がある値で減少され、最も小さな間隔が同じ値で増加される。これは、テキストラインの全長を保持し、テキストイメージにわずかな質的変更を提供するものである。本発明の他の方法と比較して、この方法の利点は、次の通りである

CODING:

```
mask off the least significant codeword bit and right-
shift the codeword
for each text line in the format file
{
  if the code bit is 1
  {
    find the longest space between words
    find the shortest space between words
    shorten longest space and lengthen shorter space
      by a chosen amount (must be <= longest space-
      shortest space)
    store text-line number, altered space
    positions, and codebit
    if codeword = 0, break
    else mask next codeword bit and
    right-shift
  }
}
```

【数4】

DECODING:

```
read in list of codeword bits and corresponding line
numbers
      and space locations in text-lines
      where coding has been performed on original
      image
set codedImage = 1
for each text-line in coded formatted file and original
formatted file
{
  if codeword bit for a text-line is 1
  then
    if coded spaces in coded image are not
    different from corresponding
    spaces in original image
    then codedImage = 0, break
}
if codedImage = 1, then image matches code
else if codedImage = 0, then image does not match code
```

【0025】

(参考)

コード化

フォーマットファイルにおける各テキストラインごとに
最小の意味のあるコードワードをマスクし、そのコードワードを右にシフト
する

```
{
  コードビットが1である場合、
  {
    ワード間の最も長い間隔を見つけ出す
    ワード間の最も短い間隔を見つけ出す
    選択された量 (最も長い間隔-最も短い間隔
    以下でなければならない) で
    最も長い間隔を短くし、最も短い間隔を長くする
    テキストラインの番号、変更した間隔位置
    およびコードビットを記憶
```

14

る。イメージをコード化するための方法で最も視覚的にわかりやすい方法のひとつである。このコードは、オリジナルなしでは解読を行うことができない。欠点は、次の通りである。このコードは、オリジナルなしでは解読を行うことができない。疑似コードは、以下の通りである。

【数3】

15

コードワード=1である場合、ブレーク
もしくは、次のコードワードビットをマスクし、
右にシフトする。

}

}

【0026】

解説：

コードワードビットのリストおよび対応するライン番号
およびテキストラインにおける間隔位置を読み取る
ここでは、コード化は、オリジナルイメージ上で行われる
コード化されたフォーマットファイルおよびオリジナルフォーマット
ファイルにおいて各テキストラインごとに

コードイメージ=1に設定

{

テキストラインのコードワードビットが1である場合、

そして、

コード化されたイメージのコード化された間隔が
オリジナルのイメージにおける対応する間隔と違いがない場合、
コードイメージ=0である、ブレーク

}

コードイメージ=1の場合、イメージは、コードと一致する、

もしくは、コードイメージ=0の場合、イメージは、コードと一致しない

【0027】2. 4 変更技法の説明的概観

図7に、ラインシフトのコード化の一例を示した。第2
のライン83は、第1のライン81からデルタと等しい
約1/150インチ下にシフトされている。異なるコード
化のため、これは、第1のライン81と第2のライン
83との間の間隔を通常よりも大きく、第2のライン8
3と第3のライン85との間の間隔を通常よりも小さく
する。第3のライン85と、第4のライン87および第
5のライン89との間隔は、通常である。

【0028】図8および図9に、ワードシフトコード化
を図示した。図8に、縦のライン91、93、95、9
7、99、101、103、105および107を示した。
これらのラインは、図8の上と下のラインの各ワード
の位置決めを示すための縦のガイドラインである。

「for」というワードは、故意にシフトされており、
したがって、テキストの下のラインには縦のライン99
において、またテキストの上のラインには縦のライン1
01が対向して置かれている。図9は、図8に示された
ものと同じテキストを示しているが、シフトされていな
いワードおよびシフトされたワードのどちらも訓練を受
けていない目には自然に見えることを実証するために、
縦のラインのないものである。

【0029】図10、11および12は、特徴強調コー
ド化を図示したものである。図10は、変更されておら
ず、代表的に標準および基準のドキュメントを表わすテ
キストの文字を示している。図11は、特徴強調を付加
した同様のドキュメントを示す。たとえば、「1」が他
の文字と同様に最初のラインの文字「t」、「l」およ

16

び「d」が有するように縦に延ばされている。図12
は、図11と同様の特徴強調を示しているが、強調を単
に強化するように誇張を付けている。図11および12
に基づく、強調した特徴のドキュメントのための適切な
コードワードは、5435デシマルであろう。

【0030】3. 0 エラー修正の応用

復元されたドキュメントに発生するノイズにより、認識
プロセスは、誤りがちになる。(エラー修正コードに基
づいた)ラインの間隔を開けるために用いられたコード
ワードの設定の賢い選択は、検査エラーの機会を最小限
にするために用いられるであろう。これは、ドキュメン
トの潜在的受信者の数(すなわち、コードワードの数)
と正しい認識の見込みとの相対関係を確立するであら
う。図示するために、ラインシフト解読がノイズ除去に
よりどのように好ましく強調されるかを詳しく次に説明
する。

【0031】一般的に、本発明の方法では、ラインシフ
ト解読は、コード化されたドキュメントの(たぶん質の
低下した)ビットマップの表示からコードワードを引き
出す(復元された、修正されていないフォーマットされ
たドキュメントファイルを解読することは、普通のこと
である)。コード化されたドキュメントの違法コピー
は、電子フォームもしくはペーパーフォームのどちらか
に復元される。ペーパーに復元された場合、ドキュメン
トのページ(1ページもしくは複数ページ)は、電子的
にスキャンされてページのビットマップイメージを作成
する。イメージファイルからコードを引き出すことは、
フォーマットファイルから引き出すようには簡単ではな

い。アスキー・テキストやフォーマットコマンドよりも、イメージはONおよびOFFのビットを含むので、パターン認識技法が、その中味を判断するためにまず使用される。さらに、ノイズが存在するので、ノイズを減少するためにイメージ加工技法が行われ、パターン認識の仕事さらに確実なものとする。イメージからのドキュメント解読に用いられる技法のうちのいくつかは次の通りである。

【0032】ソルト&ペッパーノイズ除去 — インクの不揃い、コピー機のノイズもしくは単なる紙面上のごみは、背景部分に黒いしみを、またテキストなどの前景部分に白いしみをイメージにつけてしまう。このノイズは、連続する工程の邪魔をするので、できるだけ取り除くのが好ましい。ドキュメントの質を保ちつつ、ソルト&ペッパーノイズを取り除くよう設計されたケーフィル(KF i l l) フィルターが用いられる。本当のテキストの特徴(たとえば、ピリオドや点)からノイズを分析し、ノイズを取り除く。これは、ノイズを取り除くことに対してテキストの特徴を残す側でそれら2つが対立したときエラーを起こす、保守的なフィルターであり、ドキュメントを明確にするために記述されたもので、工芸家により周知である。

【0033】デスキューイング(Deskewing) — ペーパードキュメントが写真複写され、スキャンされるごとに、ページの誤った向き、斜め置きのために、ページのテキストラインの向きは、水平から変化してしまう。加えて、写真複写機は、その光学装置のわずかな非直線性のため、歪みをもたらしてしまう。連続する工程を成功させるには、イメージファイル内でテキストラインが水平に戻されるようにこの斜めの角度が修正される必要がある。ドキュメントスペクトルもしくはドクルトル(docstrum)の使用によるデスキューイングのある方法は、ワード内へ文字をグループ化することにより開始し、その後テキストライン内へワードをグループ化するボトムアップ分割手続きの技法である。テキストラインの平均角度が1ページごとに測定され、ゼロでなかった(水平でなかった)場合は、イメージがゼロ斜角度まで回転される。最終の傾きを修正された(deskewed)イメージにするため2重線の書き入れを伴う、回転は、発行された文献に見られる標準的なデジタル型のイメージ加工手段である。

【0034】テキストライン位置 — デスキューイングのあと、テキストラインの位置を見つけることができる。「プロジェクションプロフィール」と呼ばれる標準のドキュメント加工技法が使用される。これは、各列に沿うONバリューのピクセルの合計である。テキストラインが水平に伸びるドキュメントには、このプロフィールは、幅が文字の高さと同じ最高点と、幅が隣り合うテキストライン間の白いスペースと同じ低点とを有する。プロフィールの最高点間の距離は、インターライン間隔

を決定する。

【0035】好ましい実施例では、本発明のラインシフトデコーダは、(ページプロフィール内の)隣り合う各々のテキストラインプロフィール間の距離を測定する。これは上述したように、隣り合うラインプロフィールのベースライン間の距離を測定するか、隣り合うラインプロフィールの中心間の距離を測定するかの2つの方法のうちの一つおよびベースラインに対応して、2つの最高点を有する特有のプロフィールを作り出す。各テキストラインのボトムに最も近いプロフィールにおける最高点は、ベースラインとなるように取られる。同様の最高点の値が隣り合うスキャンラインに現われる場合、最も大きな値のスキャンラインは、ベースラインのスキャンラインとして選択される。テキストラインの中心を正確に決定するためには、テキストラインのプロフィールがスキャンライン $y, y+1, \dots, y+w$ へ走り、ONビット/スキャンラインの各数は、 $h(y), h(y+1), \dots, h(y+w)$ であると仮定する。よって、テキストラインの中心は、

$$\frac{y h(y) + \dots + (y+w) h(y+w)}{h(y) + \dots + h(y+w)}$$

によって得られる。

【0036】測定されたインターラインの(すなわち、隣り合う中心もしくはベースライン間の)間隔は、白いスペースがテキストラインのシフトによって増加または減少されているかを決定するために用いられる。この工程は、各ラインごとの繰返され、ドキュメントのコードワードを決定し、オリジナルの受信者を固有に決定する。異なるコード化によるページ内のラインシフトを検出するための決定規則を記述する。テキストライン $i-1$ および $i+1$ はシフトされておらず、テキストライン i は、上か下にシフトされていると仮定する。間隔のあけられていないドキュメントでは、隣り合うベースライン間の距離もしくはベースラインの間隔は同じである。 S_{i-1} および S_i を、それぞれ、 $i-1$ と i の間、および i と $i+1$ の間であるとする。よって、決定規則は、

$S_{i-1} > S_i$ の場合: ライン i は、下にシフトされていると決定し、

$S_{i-1} < S_i$ の場合: ライン i は、上にシフトされていると決定し、

それ以外 : 不確定となる。

【0037】ベースライン検出決定規則 (3.2)

ベースライン間隔とは違って、オリジナルの間隔のあけられていないドキュメントにおける隣り合うテキストライン間の中心間隔は、必ずしも均一に間隔があけられていない。中心ベースの検出では、決定は、間隔のあけられたおよび間隔のあけられていないドキュメント内の中

心間隔の誤差に基づいている。さらに詳細には、 S_{i-1} および S_i を、それぞれ、間隔のあけられたドキュメントの $i-1$ と i との間、および i と $i+1$ との間であるとし、 t_{i-1} および t_i を間隔のあけられていないドキュメントの対応する中心間隔であるとする。よって、決定規則は、

$S_{i-1} - t_{i-1} > S_i - t_i$ の場合：ライン i は、下にシフトされていると決定し、
それ以外：ライン i は、上にシフトされていると決定する、
となる。

【0038】中心検出決定規則 (3.)

テキストラインが下に（上に）動かされている時に上に（下に）動かされているとデコーダが判断した場合、エラーが起こったと言われる。ベースライン検出において、第2のタイプのエラーが存在する。デコーダは、ラインが上か下かどちらに動かされているかどうか決定できない場合は、不確定である。コード化においてすべての他のラインが移動されているため、この情報は、デコーダに知らされて、間違いの警報はならない。

【0039】4. 0 実験結果

実験は2セット行われた。第1のセットは、典型的なイメージノイズであるが制限のあるなかで、違うフォントサイズおよび違うライン間隔でラインシフトコード化がどううまく機能するかを実験するために企てられた。第2のセットは、ドキュメント劣化がますますひどくなったとき、固定されたライン間隔シフトがどううまく検出できるかを発見するために設計された。両方の実験に用いられた装置は：

1. リコー FSIS 400 dpi のフラットベッド型電子スキャナ (Flat Bed Electronic Scanner)
2. アップル (Apple) レーザーライター II n t x (LaserWriter II n t x) 300 dpi のレーザープリンタ
3. ゼロックス (Xerox) 5052 の普通紙コピー機

これらのプリンタやコピー機は、オフィス環境で広く使用されているとみられる代表的な装置であるため選ばれた。使用される特別な機械は、多量に使用され、よく維持されるものとして特徴づけることができるであろう。ゼロックスおよび5052は、ゼロックス社の商標である。アップル社およびレーザーライターは、アップルコンピュータの商標である。リコーおよびFSIは、リコー社の商標である。

【0040】4. 1 さまざまなフォントサイズの実験
第1の実験セットは、タイムスロマンのフォントにおけるテキストのシングルスペースのページを使用している。このページは、異なるコードスキームを用いてコード化されている。異なるコード化では、各パラグラフの

最初のラインで開始されて、各パラグラフにおけるテキストのどの他のラインも動かされず保たれた。2つの動かないライン間の各ラインは、常に上か下かに動かされた。すなわち、各パラグラフにおいては、第1、第3、第5などのラインが動かされず、一方、第2、第4などのラインが動かされた。8、10または12ピクセルのフォントサイズを用いて、また、交互のライン（各パラグラフ内で）を上か下に1、2または3ピクセルでシフトして9回にわたる実験が行われた。プリンタが300 dpi の解像度を有しているので、各ピクセルは、1/300インチもしくは約9分の1ポイントに対応する。各コード化されたページは、レーザープリンタでプリントされて、3回コピーされた。レーザープリントされたページは、ゼロ回目のコピーとして参照され、 n 回目のコピー ($n \geq 1$ である) は、 $n-1$ 回目のコピーをコピーすることにより制作される。3回目のコピーがコードワードを検出するために解読された。すなわち、3回目のコピーは、電子的にスキャンされ、ビットマップイメージは、プロフィールを生成するために加工され、プロフィールは、テキストライン間隔（ベースラインおよび中心間隔の両方）を生成するために加工されて、コードワードがそれらの測定および規則（3. 2-3）を用いて検出された。

【0041】図13は、シングルスペースのテキストの1ページにつき、さまざまなフォントサイズの実験の結果を表わしている。フォントサイズが小さくなると、さらに多くのラインが1ページに配置され、さらに多くの情報がコード化される。ベースラインと中心の両方の方法は、少なくとも2ピクセルにおけるスペースではエラーなしで検出をした。中心方法は、また、1ピクセルにおけるスペースでもエラーはなかった。

【0042】図13に示されていないが、同じ復元したページの解読を繰返した場合でさえも、検出実行結果において有効であろうことは記すに値する。電子スキャンを行うことによって生じるある不規則においても当然有効であろう。ページが数回にわたりコピー・スキャンされた場合、異なる傾き角度が通常生じるであろう。この傾きは、それぞれの場合においてわずかに異なって修正され、検出結果を変化させてしまうであろう。この現象を図示するため、このテストケース（8ポイントのテキスト、1ピクセルのスペース）は、さらに3回再スキャンされた。最初のテキストラインの傾き角度（すなわち、角度修正される前の）は、各スキャンごとに異なった。3度の再スキャンで、次の解読結果がベースライン検出に現われた。5不確定、3不確定、1エラーおよび6不確定。不思議なことに、検出されなかった、または、エラーとなったライン間隔は、再度の試みにわたって多少変化した。これは、複数回1ページをスキャンすることにより得られて、結果を結合する（たとえば、平均をとる）いくつかの解読実行があることを示唆する。

【0043】4.2 普通紙コピーの実験

第2セットの実験においては、テキストのシングルスペースのページが異なるコード化方法を用いてコード化された。フォントは、タイムスロマンに、フォントサイズは、10ポイントに、また、コード化ラインシフトは、1ピクセルに設定された。そして、ページのコピーが繰返され（ゼロ回、1回、……10回）、各コピーは、別々の実験に用いられた。したがって、各連続する実験は、同じテキストページのわずかに質の落ちたものを用いた。実験の結果は、図14に一覧表にしてある。

【0044】中心検出を用いて10回目の反復したコピーでエラーが見られなかった。さらに驚くべきことは、ノイズ「マージン」に対する半分以下の有効な信号が10回目のコピーによって使い尽くされたことである。これは、数多くのコピーは、そのドキュメントが違法であるというようなたった一つのエラーでさえも生成されることが要求されるであろうことを示唆する。図14は、ベースラインの解読において、検出エラーおよび不確実性はコピーの数にともなって単純に増加しない。さらに、正確に検出されなかったライン間隔は、コピーからコピーで多少変化した。これは、ライン間隔「情報」がテキストベースラインにまだ存在し、ある付加的加工で可能となるであろうことを示唆する。

【0045】図14の結果は、我々の特徴づけスキームのコード化されていないエラー実行を報告している。しかし、実験で用いられた21ラインシフトは、勝手に選択されたものではない。コードワードは、ハミングブロックコード、1エラー修正コードから選択された3つの連結したコードワードからなる。したがって、だいたい各1/3ページは、1つのエラーから保護されている。ベースライン解読法によるすべてではないが多くのエラーおよび不確実性は、このコード化により修正されたであろう。しかしながら、コード化されていない中心検出が首尾よく行われたので、エラー修正を補足する必要があるかいは、明らかでない。

【0046】5.0 イメージの欠陥の説明および意味
普通紙コピーにより生じるイメージの欠陥は、読者にとってすべて馴染みのことである。検出結果にもっとも影響する欠陥を簡単に説明する。この説明は、イメージの欠陥のかなり質的な説明であり、それらの物質的根拠は、この概略の範囲を超えるものである。

【0047】我々が直面した主要な厄介な欠陥は、テキストラインの傾き、もしくは1点を中心としたテキストラインの回転であった。多くの実験で、我々は、 $[-3^\circ, +3^\circ]$ の間の角度の傾きを観察した。テキストラインの傾きは、いくらかの歪みの発生を犠牲にするが、イメージの回転により主に取り除かれる。汚れもまた、コピーの回数にともなって増加し、実に10回目のコピーでは、かろうじて読みうる程度になってしまう。汚れは、驚くほど検出実行において小さな意味を有するよう

である。コピー機の通常の「コピー明暗」にセットされて普通紙コピーが行われ、汚れは、コピーの暗さで増加する。コピーの回数が増加すると、明暗は一般的にページ上で変化し、かなり薄れたところが見られた。汚れや薄れが解読作業にとってさらに不利益であるかどうかは、定かでない。コピーサイズの拡大または縮小は、別の可能性のある問題である。10回のコピー後に、ページの縦または横の4%の変化を発見することは稀なことではない。さらに、ページの縦および横の拡大は、著しく異なるものとなりうる。コピーサイズの変化により、別のコード化、すなわち、隣り合うテキストライン間の絶対シフトと比較するコード化情報を使用しなければならなかった。コピーの簡単な検査は、水平方向および垂直方向の両方のずれおよび他のあまり重要でないイメージの欠陥（たとえば、ソルト&ペッパーノイズ）を示す。おそらく、最も驚くべき質の低下は、「ベースラインのうねり」（すなわち、テキストラインの非一定の傾き）である。検出が、この特別なイメージの質の低下によって劇的に影響を受けないことは、驚くべきことである。

【0048】5.1 分析ノイズモデル

この小区分では、テキストライン中心へ影響するノイズの簡単なモデルを提供する。第一のタイプのノイズモデルは、ドキュメントをプリントする際の歪み、第2のタイプのノイズモデルは、コピーの際の歪みである。この第2のタイプのノイズは、コピーの回数に伴って増加する。一方、第1のタイプのノイズは、増加しない。ノイズの蓄積が、図15に示される。これは、ドキュメントは、エンコーダ201からオリジナルプリンタ203、第1のコピー機、第2のコピー機、一連のコピー機の最後、すなわち、コピー機209、その後デコーダ211を移動するという理論モデルを図示している。

【0049】 $n+1$ 個のテキストラインを有するテキストの1ページは、いわばトップのページマージンから測定して、テキストラインの中心を示す $n+1$ 個の縦軸 y_1, \dots, y_{n+1} を生じる。隣り合う中心間のスキャンラインにおける中心間隔もしくは距離は、

$$t_i = y_i + 1 - y_i \quad i = 1, \dots, n$$

である。したがって、ラインシフトの検出には、 $n+1$ 個のテキストラインのページは、 n の中心スペースにより効果的に記述される。図15では、 i 番目のラインスペースのシフト C_i は、余分なスペースが加えられていた場合は、正であり、スペースが減じられていた場合は、負である。プリンタのノイズ、 V_1 、モデルは、プリント、スキャンおよびイメージ加工による歪みの（ i 番目の中心スペースにおける）累積的な効果である。 j 番目のコピーの作成には、 i 番目の中心スペースに任意のノイズ N_{ij} を加える。 K 番目のコピー後のデコーダの入力において、オリジナルの中心スペース $t_1 + c_i$ は、 s_{ik} に変形された。プリント、スキャンおよびイメージ

加工の物質的プロセスはコピーとは関係ないので、任意の変数 v_i ($i=1, \dots, n$) は、 N_{ji} ($i=1, \dots, n, j=1, \dots, K$) と関係ない。

$$S_i = t_i + c_i + V_i \quad i=1, \dots, n \quad (4.1)$$

に変形し、 V_i ($i=1, \dots, n$) は、独立しており、同様にガウス・ランダム変数が分配される。この前提は、 μ の平均は $=0.0528$ ピクセルであり、 $\delta 12$ の変動は 0.140 であるという測定により成り立つ。

【0051】次に、ノイズの影響がコピーにより生じた※

$$S_{i1} = S_i + N_{i1} \quad i=1, \dots, n \quad (4.2)$$

であり、 N_{i1} は、 i 番目の中心スペース S_i における、コピーの際の傾き、スケールングおよび他の写真複写の歪みの累積的な影響を要約する任意のノイズである。★

$$S_{ij} = S_{ij-1} + N_{ij} \quad i=1, \dots, n \quad (4.3)$$

であり、 N_{ij} は、 $j-1$ 回目のコピーをコピーすることにより生じるノイズである。したがって、中心スペース☆

$$S_{ij} = S_i + (N_{i1} + \dots + N_{ij}) \quad (4.4)$$

【0052】この取られた測定は、任意のコピー機における以外に単純な統計反応を示唆する。ノイズ要素 N_{ij} ($j=1, \dots, K$) は、 μ の平均 $=0.0666$ ピクセルと、 $\delta 2$ の変動 $=0.017$ ピクセル² を有するガ◆

$$S_{ij} = S_i + \eta_{ij} \quad i=1, \dots, n \quad (4.2)$$

であり、 η_{ij} は、 $j\mu$ と、 $j\delta 2$ を有するガウス曲線である。

【0053】プリンタノイズおよびコピー機ノイズは、中心検出下で可能なエラーを概算するため合わせられる。3つの隣り合う異なってコード化されたテキストラインは、ライン $i-1$ および $i+1$ がシフトされず、ライン i が c ピクセルで (上もしくは下に) シフトされるように表示されたと仮定する。 t_{i-1} および t_i は、オリジナルのスペースの空けられていないドキュメントにおけるそれらのライン間の中心スペースとし、 S_{i-1} およ *

$$S_{ji-1} = t_{i-1} + c + V_{i-1} + \eta_{ji-1} \quad (4.8)$$

$$S_{ij} = t_i - c + V_i + \eta_{ij} \quad (4.9)$$

であり、 η_{ij} は、(4.5) で画定されるものである。

【0054】 j 回目のコピーが復元され、解読されると※

$$V_{i-1} - V_i > \eta_{ij} - \eta_{ji-1} - 2c \text{ の場合: ラインは下にシフトされていると決定} \quad (4.10)$$

それ以外 : ラインは上にシフトされていると決定。
ランダム変数 V_{i-1} 、 V_i および η_{ji-1} 、 η_{ij} は、互いに独立しているので、判断変数 $D \Delta (V_{i-1} - V_i) + (\eta_{ji-1} - \eta_{ij})$ は、ゼロ平均および変動 $2(\delta 12 + j\delta 2)$ をともなうガウス曲線である。したがって、与えられたラインがエラーと解読される可能性は、

$$p(D > 2c \text{ 上にシフト}) = 1/2 p(D \leq -2c \text{ 下にシフト}) = 1/2 p(D-2) \quad (4.11)$$

エラーの可能性は、補足のエラー関数を用いて容易に評価される。 $\delta 12 = 0.140$ および $\delta 2 = 0.017$ 測定を用いると、エラーの可能性は、20回目のコピーでたったの約2%である。

【0055】5.2 ベースラインおよび中心検出の比較アルゴリズム

テキストラインプロフィールのベースラインもしくは中

*【0050】 $n+1$ 個のテキストラインのページが中心スペース t_1, \dots, t_n で描写されるとしよう。プリンターのノイズは、それらのスペースを

※と仮定する。 $n+1$ 個のテキストラインのページの0回目のコピーが、中心スペース s_1, \dots, s_n を有すると仮定する。ページの1回目のコピーは、中心スペース s_{i1}, \dots, s_{n1} により表わされ、

★番目のコピー ($j \geq 1$) 後、中心スペースは、 s_{ij}, \dots, s_{nj} により表示される。(4.2) のように、それらの中心スペースは、

☆ S_{ij} は、全ノイズにより汚される。

◆ウス・ランダム変数により首尾よくモデル化される。測定は、ランダム変数 N_{i1}, \dots, N_{ij} 汚され、正常な状態では、それらは、独立している。したがって、 j 回目のコピーにおける中心スペース S_{ij} は、

* $b_i S_{i1}$ は、コード化されたドキュメントのゼロ回目のコピー上の対応するスペースであるとする。よって、

$$S_{i-1} = t_{i-1} + c + V_{i-1} \quad (4.6)$$

$$S_i = t_i - c + V_i \quad (4.7)$$

であり、ライン i が下にシフトされている場合は、 $c = +1$ であり、ライン i が上にシフトされている場合は、 $c = -1$ である。 S_{ji-1} および S_{ij} は、ドキュメントの j 回目のコピー上の対応する中心スペースであるとする。よって、

※仮定する。上記の (4.8) および (4.9) を検出規則 (3.3) に加える。

心のどちらかを用いる検出は、異なった利点および欠点を提供する。想像するように、実験結果は、中心による検出は、小さなラインシフト (すなわち、1ピクセル) で、且つ大きな歪みを受けやすいコード化したページにおいてはベースラインによる検出よりも優れていることを示している。中心位置は平均となり、真の評価値となるが、ベースライン位置が完全数とされるためこの機能

の違いは、大きくなる。ベースライン位置はテキストラインプロフィールの最高点の検出により決定されることを思い出してほしい。この最高点は、時々はっきりせず、ベースラインに隣接するスキャンライン上のプロフィール値がしばしば最高点の値に近くなる。よって、比較的小さなノイズは、隣接するスキャンラインへシフトするよう最高点を生じる。テキストラインが1ピクセルシフトでコード化された場合、一本のスキャンラインシフトは、検出エラーを起こすに十分である。

【0056】ベースラインよりも、中心は、そのようなイメージの欠陥を被りにくい。ベースラインは、比較的、ラインの傾き（もしくは、さらに正確には、傾き修正により生じたノイズ）の被害を受けやすい。中心検出は、ベースライン検出より、優れているが、ベースラインは、他の利点を有している。特に、コード化されたドキュメントをオリジナルのスペースの空けられていないドキュメントなしで解読することができる。安全なドキュメント配給者は、解読のため中心スペースのオリジナルドキュメントのライブラリを管理する必要から解放される。最後に、特に確実な、低いエラーの可能性の検出スキームを提供するためには、両方の検出技法を（また、実際は他の技法と）結合して用いることができる。

【0057】6.0 結論

オリジナルのコピーがそれぞれ特有なものである場合、ドキュメントの違法コピーの作成および配布を防ぐことができ、また、特定の受信者と関連づけることができる。ドキュメントを特有のものにする幾つかの技法が記述された。それらの技法のうちのひとつは、テキストラインシフトに基づいたものであり、普通の読者には識別できないように小さなラインスペースの変更は、何度かコピーされた後であっても、ドキュメントのコピーから復元できることを実証するため実験として履行された。

【0058】実験において、各パラグラフ内の奇数ラインの位置はそのまま、一方、偶数ラインは、わずかに上もしくは下に移動される。異なるラインシフトを選択することにより、情報は、ドキュメントの中にコード化される。ドキュメントが実験中電子フォームのままの場合、コード化された情報を取り戻すことは通常のことである。ペーパーコピーから情報を取り戻すには、ドキュメントをスキャンしてコンピューターに戻す。2つの検出方法が考慮された。ひとつは、各ラインの文字の低点の位置に基づき、他方は、各ラインのかたまりの中心に基づき。ベースラインを用いる利点は、それらはコード化する前等しくスペースが空けられており、母型を参照することなしに情報を引き出すことができる。ラインのかたまりの中心は、等しくスペースが空けられていない、しかしながら、この技法は、プリントおよびコピーの工程で生じたさまざまな歪みに対してさらに復元力のあるものであることが発見された。

【0059】生じたさまざまな歪みが隣接するライン間の誤差を考慮するとき取り消されたので、この異なったコード化機構が選択された。実験では、ドキュメントのラインは、1/300インチほどの小ささで上もしくは下に動かされ、ドキュメントは、10回ほどの多さでコピーされ、その後、ドキュメントはスキャンされてコンピュータに入れられる。行われた実験では、中心解読機構が、限りなく小さなエラー率をもたらした。上述の技法を鑑みて本発明のさまざまな修正および変更が可能であることは、明らかである。したがって、添付クレームの範囲内において、ここで特別に記述したもの以外で、本発明を実行できることは理解されることである。

【0060】

【発明の効果】上述したような本発明は、各加入者特有の独特な識別コードを作成することにより、電子発行されたドキュメントの違法コピーを防止でき、発行者および著作権者が違法コピーを分析できるようにし、その元の加入者を判断することができる。

【図面の簡単な説明】

【図1】本発明の方法の好ましい実施例のフローチャートである。

【図2】本発明の方法におけるコード化実行のフローチャートである。

【図3】本発明の方法における解読実行のフローチャートである。

【図4】ポストスクリプトファイルのための一行間のコード化実行のための疑似コードを示す説明図である。

【図5】テキストラインシフトコード化を用いた回収されたドキュメントのプロフィールを示す説明図である。

【図6】5×5ピクセル配列において強調する特徴の3つの例を示す説明図である。

【図7】定性的に示した行間測定を用いたラインシフトコード化を示す説明図である。

【図8】通常のおよびシフトワードの行間を強調するため縦のラインでのワードシフトコード化を示す説明図である。

【図9】縦のライン以外は図8に示されたものと同一のテキストを示し、シフトされていないおよびシフトされた両方のワード間隔が訓練を受けていない目にとって自然に現われることの説明図である。

【図10】特徴を強調しないドキュメントのテキストの一例を示す説明図である。

【図11】特徴を強調した図10のテキストを示す説明図である。

【図12】誇張により強調された同一の特徴を有する図11のテキストを示す説明図である。

【図13】行間とフォントサイズとが変更されるときベースラインと中心検出結果との比較を示す説明図である。

【図14】テキストのページが反復的にコピーされると

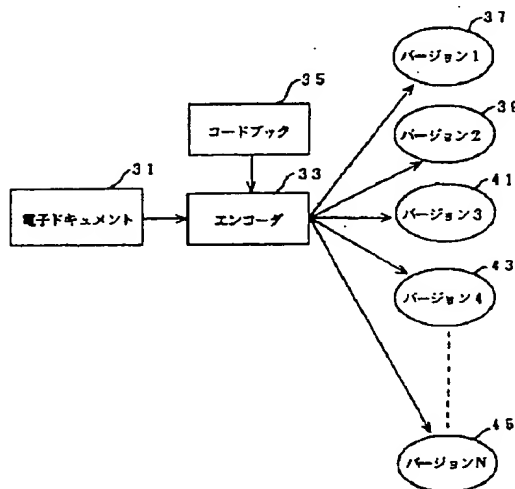
きのベースラインとと中心検出との比較を示す説明図である。

【図15】ノイズ蓄積モデルのための概略説明図である。

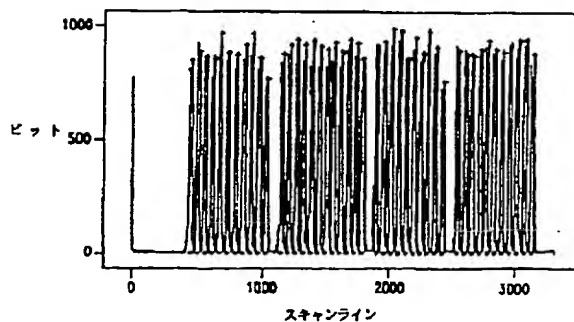
【符号の説明】

- 3 発行者
- 9 エンコーダ
- 11 ペーパーコピー配布チャネル
- 13 電子配布チャネル
- 15 ユーザーディスプレイ
- 17 ユーザープリンタ
- 21 ドキュメントデータベース
- 27 普通紙コピー機
- 29 違法のペーパーコピー
- 31 電子ドキュメント
- 33 エンコーダ

【図2】

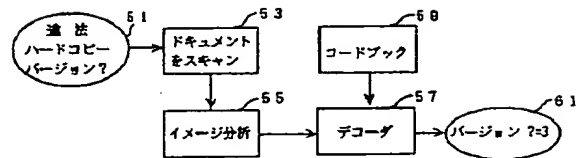


【図5】

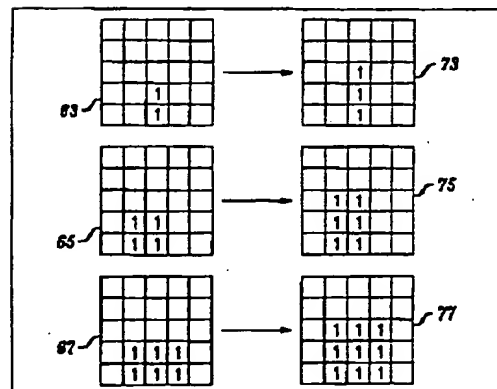


- 35 コードブック
- 63、65、67 通常のコード化
- 73、75、77 強調コード化
- 81 第1のライン
- 83 第2のライン
- 85 第3のライン
- 87 第4のライン
- 89 第5のライン
- 91、93、95、97、99、101、103、105、107 縦のライン
- 201 エンコーダ
- 203 オリジナルプリンタ
- 205 第1のコピー機
- 207 第2のコピー機
- 209 最後のコピー機K

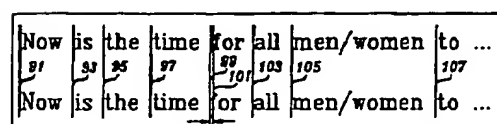
【図3】



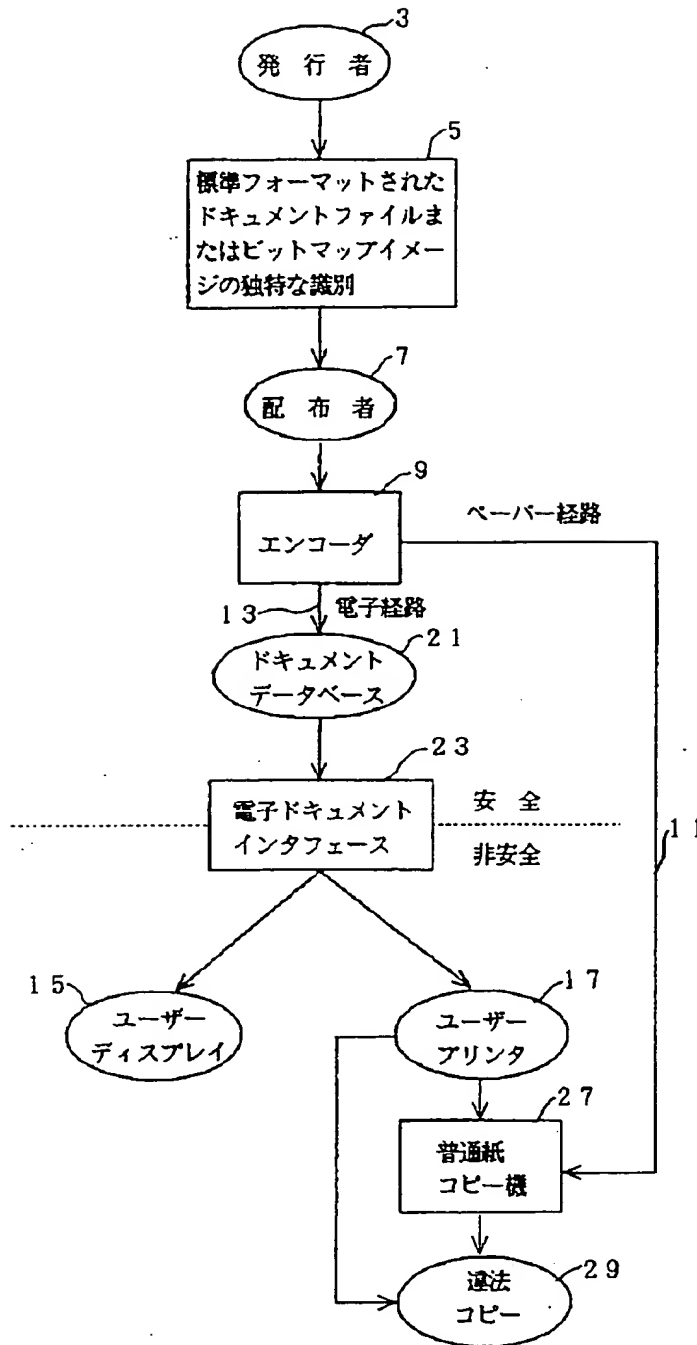
【図6】



【図8】



【図 1】



【図 10】

IS AND HEORY	1	Incremental Mod by John H. Mulla
STEMS	20	A Unified Frame by J. A. Brzozow

【図 11】

IS AND HEORY	1	Incremental Mod by John H. Mulla
STEMS	20	A Unified Frame by J. A. Brzozow

【図 14】

コピー #	ベースライン検出結果	中心検出結果
3	20 正しい, 1 不確定	21/21 正しい
4	19 正しい, 1 エラー, 1 不確定	21/21 正しい
5	21 正しい	21/21 正しい
6	19 正しい, 1 エラー, 1 不確定	21/21 正しい
7	18 正しい, 1 エラー, 2 不確定	21/21 正しい
8	16 正しい, 1 エラー, 4 不確定	21/21 正しい
9	19 正しい, 2 エラー	21/21 正しい
10	19 正しい, 1 エラー, 1 不確定	21/21 正しい

【図4】

```

##
## ポストスクリプトファイルにおけるラインスペースをコード化
##

codeword[] = 0, 1, 0, 0, -1, 0, -1, 0, 1, 0, 0      #コードワード
spacing = 2                                           #ピクセルにおける超過スペース

open original_postscript_file
  get (page) {                                       # 次のページを得る
    line_count = 0
    get (line) {                                    # 次のラインを得る
      linecount = linecount + 1                    # ライン数のトラックを保持
      y_coordinate = y_coordinate + (codeword[linecount] * spacing)
                                                    # 必要であればライン位置の変更
                                                    # ポストスクリプトファイルを修正するため
                                                    # ラインを書き込み
    }
  }
  write (line)
}

```

【図7】

This is a method of altering a document by vertically shifting the location of text lines to uniquely encode the document. The encoding is most easily applied to the format file. The embedded code word may be decoded from the format file or bitmap.

【図9】

Now is the time for all men/women to ...
 Now is the time for all men/women to ...

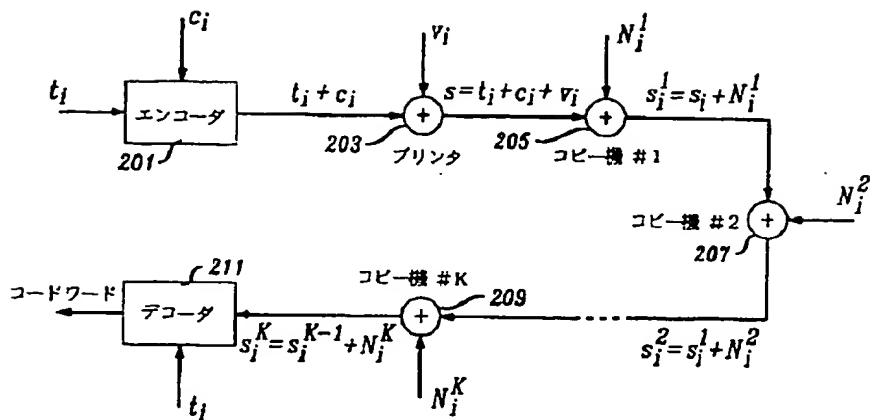
【図12】

IS AND	1	Incremental Mod
HEORY		by John H. Muller
STEMS	20	A Unified Frame
		by J. A. Brzozow

【図13】

フォントサイズ (ポイント)	ラインシフト (ピクセル)	ベースライン検出結果	中心検出結果
8	1	18/23 正しい, 5 不明	21/23 正しい
	2	23/23 正しい	23/23 正しい
	3	23/23 正しい	23/23 正しい
10	1	21/23 正しい	21/21 正しい
	2	22/21 正しい	22/21 正しい
	3	21/21 正しい	21/21 正しい
12	1	18/19 正しい, 1 エラー	18/19 正しい
	2	19/19 正しい	19/19 正しい
	3	19/19 正しい	19/19 正しい

【図 15】



フロントページの続き

(72) 発明者 スチーヴン エッチ. ロウ
 アメリカ合衆国, 08807 ニュージャージー,
 ブリッジウォーター, フィンダーン
 アヴェニュー 84-エー

(72) 発明者 ニコラス エフ. マクゼムチュック
 アメリカ合衆国, 07092 ニュージャージー,
 マウンテンサイド, ローリング ロック
 ロード 355

(72) 発明者 ローレンス ピー. オーゴーマン
 アメリカ合衆国, 07940 ニュージャージー,
 マディソン, アルブライト サークル

レイアウト構造を利用したページ記述への電子透かし埋め込み手法

天野富夫, 平山唯樹

E-mail: amano@jp.ibm.com, hirayama@jp.ibm.com

日本アイ・ビー・エム東京基礎研究所

あらまし

デジタルライブラリーや情報配信サービスにおいて流通するコンテンツの大部分は文書であり、文書データへの電子透かし埋め込み技術への需要は大きい。本論文では文字の間隔を変化させて文書のページ記述に透かしを埋め込む手法について報告する。提案手法では、ページ記述内の文字オブジェクトをそのレイアウト構造上で順序付けし、オリジナル文書を必要としない検出や統計的手法の適用を可能にしている。また、文字間隔を比較するさいにはフォントのサイズや各文字の縦横比を考慮した正規化を行ない、使用フォントやテキストの内容の異なる文書に対して同じように埋め込み/検出ができるようにしている。代表的なページ記述であるPDF(Portable Document Format)を対象として本手法をインプリメントした。処理できるオペレータやフォントの種類は限られているものの、本手法の実現可能性を確認することができた。

A method for embedding digital watermark in page descriptions

Tomio Amano and Yuki Hirayama

IBM Japan, Ltd., Tokyo Research Laboratory

Abstract

This paper proposes a method for embedding/detecting a watermark in document page descriptions such as PDF (Portable Document Format). The method uses normalized intervals between two succeeding characters. A watermark can be encoded as slight changes in these intervals. A page layout structure is constructed from the page description and used to order character objects; so that a watermark can be detected without original data. We have implemented a embedding/detecting program which can deal with small set of PDF operators, and investigated the feasibility of our method.

1 はじめに

インターネットやCD-ROMの普及によってデジタルコンテンツを容易に配布、流通させることが可能になった。このことは、インターネットを利用した情報配信サービスなどの新たなアプリケーション/ビジネスの発達を促す反面、コンテンツの不正な複製や改ざんが行われる危険性も増大させた。不正行為を防止する手段としてコンテンツデータ自体に権利等の情報を埋め込む電子透かし技術が注目されている。

電子透かしはコンテンツデータと非可分なかたちで埋め込まれるため通常の流通過程において消去されることはない。コンテンツから透かし情報を検出することにより、著作権者や流通経路(誰が複製を行なったのか)を同定できるため、デジタルコンテンツの不正利用を抑止する効果があると期待される。また、正しい透かしが埋め込まれているか否かによってコンテンツが改ざんされていないことを他者に保証する[1]、レーティング情報を埋め込んでおいてブラウザー/プロキシ側でスクリーニングを行なう(例えば成人向けコンテンツの表示を制限する)、等の利用法も検討されている。

デジタルライブラリーや情報配信サービスにおいて流通するコンテンツの大部分は文書データであり、静止画や動画と並んで文書データへの透かし埋め込み技術が必要である。著者らは特に、文書の「見えかた」を規定するページ記述への透かし埋め込みが重要だと考えている。多種多様なフォーマットで作成された文書がページ記述に変換された後、配布される場合が多いからである。テキストのコード列自体は冗長度が低く透かしの埋め込みは困難であるがページ記述には文字の大きさや位置等のデータが含まれるためそれらの冗長性を利用して透かし情報を埋め込むことが可能である。

本論文では、代表的なページ記述フォーマットであるAdobe社のPDF(Portable Document Format[2])に透かしを埋め込む手法について報告する。PDFはプリンタドライバを経由することによりほとんど全ての文書フォーマットから変換することができ、閲覧用ソフトウェアも無料で提供されている。そのためインターネットやCD-ROMによるページ記述文書の配布形態として広く用いられている。PDFのかたちで書籍や文書情報を販売する電子商店も既に存在する。したがって、ページ記述への透かし埋め込みの具体的な対象としてPDFへの埋め込み/検出の実現可能性を検証することの意義は大きいと考える。

本論文では、まず2節でPDFへの透かし埋め込みに関する要件を整理し従来手法を概観する。3節では提案埋め込み手法を、埋め込みの基本単位、その順序付けの方法、統計的手法を用いたビットの埋め込み/検出の順で説明する。4節で実際の埋め込みの例をしめし、実際の文書における特徴の変化の検出方法の妥当性を検証する。5節では、求められている要件に提案手法がどの程度対処しているか考察を行い、6節でまとめと将来の展望を述べる。

2 ページ記述への電子透かしの要件

不可視な電子透かしが満たすべき基本的な要件としては、埋め込みによってコンテンツの外観が不自然にならないこと、アプリケーションが必要とするだけの容量の情報を埋め込めること、が挙げられる。さらに実際の運用を考えた場合には、以下の3つの要件も考慮する必要がある。

(1) オリジナル文書無しでも透かしの検出が可能: 透かしはコンテンツ内のなんらかの特徴値の変化として記述される。この変化を見つける最も簡単な方法は埋め込み前のオリジナルコンテンツと検出対象を比較することである。しかし、この方式では透かしを埋め込んだ側で将来の検出のために全てのオリジナル文書を保持している必要があり、電子透かしシステムの運用上大きな負担となる。適当な「キー」を所有していれば複数の文書データに対して検出処理が行える埋め込み方式が望ましい。

(2) 日常的な操作に対する耐性: 透かしは流通過程でうける通常の操作に対して頑健である必要がある。PDFに対するこの種の操作としてはページ単位での抜き出しやフォーマット変換(PDF→PostScript→PDF)がある。コメントやアノテーションの機能を利用した透かしはフォーマット変換により消去されてしまうため、より文書データの内容に結びついたかたちで透かしを埋め込む必要がある。改ざん防止に透かしを用いる場合はこのレベルの耐性があれば利用可能である。

(3) 意図的な攻撃に対する耐性: 著作権管理等のアプリケーションに用いる場合には意図的に透かしを消去・破壊しようとする攻撃に対しても頑健である必要がある。

ページ記述への電子透かし埋め込みについては既にいくつかの提案が行なわれている。渋谷ら[3]はPostScriptやPDFの表現の多様性に着目し透かしを埋め込むことを提案しているが、当該論文では具体的

なインプリメントまでは行っていない。特許公報[4, 5]にはページ記述内の文字や行の位置情報を利用して透かしを埋め込む手法が開示されている。しかし、上記の方法は[5]の文字行のベースラインを操作する方法を除いて、透かしの検出に関してはオリジナル文書と比較する方式を想定している。

3 PDFへの透かし埋め込み手法

本節では、埋め込みプリミティブとプリミティブの順序付け[6]という二つの観点からPDFへの透かし埋め込み手法を概説する。提案手法では、埋め込みプリミティブとして連続する二つの文字の間隔を用い順序付けの方法として各文字をページのレイアウト構造にマッピングしたときの位置(何行目の何文字目か)を利用している。順序付けを行うことによって、オリジナル文書を必要としない透かし検出、統計的手法の導入による頑健性の獲得などのメリットが得られる。

3.1 埋め込みプリミティブ

埋め込みプリミティブとは透かし埋め込み処理のためになんらかの特徴値を変化させる単位である。ページ記述内には文書の外観を規定する文字やフォント、線分などのオブジェクトが含まれているが、われわれは連続する2文字間の間隔ある文字の左端から次の文字の左端までの水平距離、を埋め込みプリミティブとして用いている。

水平距離の値はフォントの種類やサイズに影響され、また文字コードによっても変化する(例えば“f”のような細長い文字が並んでいる場合と“m”や“w”が並んでいる場合とで値は異なってくる)。使用フォントや内容の異なる文書間でも同一の基準で比較ができるよう、埋め込みおよび検出にはフォントごと文字ごとの標準的な幅で正規化した間隔(正規化間隔と呼ぶことにする)を用いている。

文字は一般文書のページ記述において最も多く出現するオブジェクトであり、文字行や線分を埋め込みプリミティブに使う場合に比較して多くの情報を埋め込むことができる。文字(列)の位置の記述は多くのページ記述フォーマット中に存在するため、PDF以外への埋め込みにも適用が期待できる。

3.2 プリミティブの順序付け

埋め込みプリミティブの集合を用いて透かしをコード化するためには、プリミティブを順序付けする適当な方法が必要である。透かし検出時にあるプリミティブの変化が観測されたとき、どんな意味があるのか解釈するためにはそれが埋め込み時に使われたプリミティブ集合のどの要素なのか識別しなくてはならないからである。オリジナル文書と比較するということは、ファイル内でのオブジェクトの出現順によってプリミティブを識別し「 n 番目の文字と次の文字の間がひらいてからビット1が埋め込まれている」といった判定を行なうことを意味する。しかし、ファイル内での出現順による識別は文書データごとに固有のものとなってしまう。前節で述べたようなオリジナル文書の管理という運用上の問題が発生する。さらに、同じ外観が得られるならばファイル中におけるオブジェクトの記述の順番は任意でよい。フォーマット変換や透かしへの攻撃によってその順番が変更されるかもしれない。埋め込みを行なうためには同じ外観を与える複数のページ記述に対してもオブジェクトの順序が一意に決められるようにしておく必要がある。

ファイル内での出現順にかかわらず一意に決まる順序付けとして、筆者らは文字→行→段組→ページといったレイアウト構造上での順序(何行目の何文字目か)を用いることにした。このようなレイアウトの論理構造はページ記述には含まれていないが、文書画像の解析技術(例えば[7])を使えば文字オブジェクトの物理的な座標情報から再構築することができる。流通過程で個々の文字オブジェクトの出現順が変化してもレイアウト構造は変化しないためそれに基づくプリミティブの順序付けも影響を受けない。

3.3 埋め込みアルゴリズム

図1に、提案手法に基づく透かし埋め込み/検出の流れをしめす。埋め込みにあたっては、まずPDFファイル中の文字オブジェクト集合を解析して対象ページのレイアウト構造を構築する。構築されたレイアウトは人間が理解するように正しいものである必要はなく、埋め込み時と検出時で同じ結果が得られればよい。続いてあるキー(シード)に基づいて発生させた乱数列によって透かしの各ビットを埋め込む位置を決定する。この位置はレイアウト構造上で、 x 段目の y 行目 z 文字目といった値で表現される。そこにある文字オブジェクトを動かして正規化間隔を変化させることにより埋

め込みを行なう。透かしの検出にあたっては、まず埋め込みと同様にレイアウト構造を構築する。続いて埋め込みに用いたのと同じシードから発生させた乱数-埋め込みプリミティブとして操作された文字オブジェクトの場所をしめすことになる-から得られた文字と次の文字の間の間隔を調べるによりビットごとの埋め込み判定を行なう。

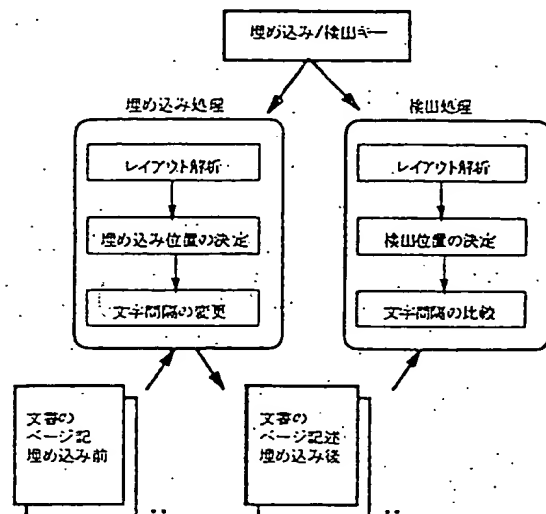


図 1: PDF への透かし埋め込み/検出処理フロー。

3.1で述べたような正規化を行っても、行の右端をそろえる処理、あるいは禁則処理による文字の追い込み/追い出しが原因で埋め込みプリミティブの値が変動することがある。実際のワープロやフォーマッタの出力はこのような変動を含んでいることが多い。オリジナルとの比較なしに間隔の変化が透かしをあらわしていることを判定するためにはなんらかの推定を行わなくてはならない。ここでは、1ビットを埋め込むために複数のプリミティブを用いて統計的な推定によってビットを検出する方法 [8] を採用している。

統計的手法では次式で計算される値によって1ビットの判定を行なう。

$$d = (1/N) \sum_n (a_n - b_n) \quad (1)$$

a_n, b_n は乱数によって指定された文字列オブジェクトにおける正規化間隔の値である。1ビットを埋め込むために a_n, b_n の組を N 個用い、1を埋め込む場合には a_n を大きくし b_n を小さくする、0を埋め込む場合には a_n を小さくして b_n を大きくする、といった操作を行っておく。通常、式(1)の d は0付近の値をとるが、埋め込みが行われた組に関しては正または負の大きな絶対値の

値をしめす。この値が0から十分離れていれば閾値処理によってビットの値を検出することができる。単独の組で大きな絶対値を得られなくても N を大きくすることにより、 d の標準偏差を減少させて閾値処理の精度を高めることができる。統計的手法の利点は、検出の誤り確率と埋め込める容量の間のトレードオフをアプリケーションの要請に応じて設計できることである。

4 透かし埋め込み例

基本的な要件に関して本手法を評価するため、PDF のオペレータやフォントの種類を限定して埋め込み/検出プログラム (Java 言語で約3,300行) をインプリメントした。

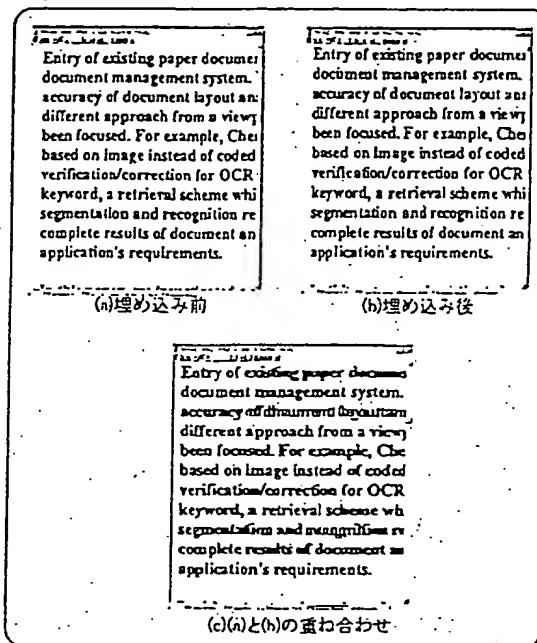


図 2: PDF への透かし埋め込み例。

埋め込みのために文字間隔1ヶ所に与えた変更は、正規化前の値で対象文字のポイント数×0.05である。このプログラムを用いて、ページの記述に“Tokyo Research Laboratory”という文字列を透かしとして埋め込んだ例を図2にしめす。この例では、12ポイントのサイズで印刷される文字を0.6ポイントずつ左右に動かすことにより透かしを埋め込んでいる。埋め込み前と後の2つの表示画面を比べてもはっきりした違いはわからないが、重ね合わせてみると文字の位置が微妙に変化していることがわかる。多種の文書に対する定量的評価を行なう必要はあるものの、埋め込み処理が「ひとめで

わかるような」不自然さをもたらさないことは確認できた。

表 1: 文書データにおける正規化間隔の変動

正規化間隔	ワープロ	間隔拡張	間隔縮小
平均	0.996	1.054	0.946
分散	0.0000835	0.0000415	0.0000361

表 2: 正規化間隔標準偏差と推定誤り率の関係

	標準偏差 無/有	比	誤り率
N=1	0.0129/0.0156	3.78	0.0156 %
N=2	0.00913/0.0110	5.35	0.00000898 %
N=3	0.00746/0.00903	6.55	0.000000006 %

禁則処理等が原因で正規化間隔が変動していても透かしの埋め込み/検出ができることを確認するため、実際にワープロで正規化間隔が変動している PDF を作成し変動の幅を透かしによる変化と比較した。ワープロ出力から生成された PDF は今回インプリメントしたプログラムでは処理できないオペレータを含んでいたため、別にプログラムを用意して正規化間隔の値のみを計算している。表 1 は、正規化間隔の平均と分散を (a) 何も埋め込まれていない (しかし正規化間隔が変動している) ワープロ出力、(b) 埋め込みのために幅を広げたデータ、(c) 埋め込みのために幅を狭くしたデータ、に関してまとめたものである。(b)、(c) の埋め込みは間隔の変動のないデータに対して行っている。表 1 の値から式 (1) の d の値の平均は埋め込み無しのデータに対して 0、有りのデータに対して ± 0.108 であることがわかる。表 2 は表 1 の分散の値から分散の性質

$$\sigma^2(a+b) = \sigma^2(a) + \sigma^2(b) \quad (2)$$

$$\sigma^2(ka) = k^2 \sigma^2(a) \quad (3)$$

より式 (1) の d の分散 (標準偏差) を計算したものである。実際の埋め込みは、(a) のようなばらつきを含むデータに対して行われるため埋め込み有りの分散の値には (a) の分散の値も加算してある。N=1,2,3 の場合に対して埋め込み有りと無しの平均値の差 0.108 (1.054-0.946) が 2 つの標準偏差の和の何倍になるかという比をしめす。さらにその比を使って埋め込み有り/無しを判定する閾値を決めたときの False positive error (埋め込みが無いのに有りと判定してしまう確率) を正規分布を仮定して計算した。この表から、埋め込み対象の正規化間隔の変動が今回調査したワープロ出力程度

であれば、N=2 とすれば誤り確率 10^{-7} 以下で埋め込み/検出が可能であると期待できる。

5 考察

2 節で述べた要件と本手法の適合性について考察する。

1 ページに埋め込める透かしの容量は、ページ内の文字数に依存する。統計的手法を用いて N=2 とした場合、1 ビットの埋め込みには 4 個所の文字間隔を変更する必要がある。A4 判 1 ページの文字数は図や写真が無い場合で 1600 から 2000 文字程度であり、単純計算で 400~500 ビットの透かしを埋め込むことができる。アノテーションを入れるといった目的には足りないが、購入者識別用の ID、連絡先 URL、デジタル署名等を埋め込むことは可能な容量である。図や写真が大きな面積を占めるページへの埋め込みに関しては、他のオブジェクトへの埋め込み手法との併用が必要である。

文字オブジェクトの物理的位置情報はページ記述のなかではテキストのコード列に次いで本質的な情報であり通常の流通過程では変更されない。この種の情報は、ページ記述一般に共通しておりフォーマット変換によっても失われる可能性は低い。実際、透かしを埋め込んだ PDF を PostScript に変換した後もう一度 PDF に変換しても透かしを検出することができる。

インプリメントしたプログラムは、埋め込み/検出の実現可能性を検証するためのもので意図的な攻撃に対処するための処理は行っていない。強化を行ってどこまで頑健にできるか、いくつかの攻撃方法について検討する。

ランダム攻撃とは文字間隔によって透かしが埋め込まれていることを知る者がランダムに (文書の外観を変化させない範囲で) 文字オブジェクトを動かしてしまいう攻撃をいう。採用した統計的手法は、この種の攻撃に耐性をもつ [8]。攻撃による幅の増減が式 (1) に加わったとき、打ち消しあってしまうからである。ただし、そのためにはある程度大きな N を使う必要があるため埋め込める容量は低下する。

結託攻撃とは、同じ文書を複数の人間が購入し比較することにより、埋め込まれた Finger Print 情報 (誰に販売したのかをしめす情報) の場所を同定し破壊する攻撃である。本手法の場合には、埋め込んだオブジェクト以外の文字オブジェクトもランダムに動かして透かしを埋め込んだ場所をわからなくする。文字オブジェクトのファイル中での出現順序を変えて (外観にはなん

の影響もでない)単純な比較ができないようにする,等の対処が考えられる.他に埋め込みの手法ではなく, ID等を符号化するとき冗長な表現を用いて符号化データの一部分が破壊されても結託ユーザーを特定できるようにする対処策[9]も検討されている.

ページ記述フォーマットの仕様が公開されている場合, 原理的にはテキスト等の内容を抜き出して(PDFには暗号化の機能があるが透かしによって管理したいのは復号された後の文書の流通である), 新たな文書として再構成することが可能である. その意味で消去不可能な透かしは存在しない. しかしながら, 透かしの消去を試みる者の目的が複製した文書を販売して不当に(安易に)利益を得ることであるとすれば, 上述の出現順の変更などで透かしの消去に要するコストを彼(彼女)らが得るであろう利益に対して相対的に大きくすることにより一定の抑止効果が得られると期待できる.

6 まとめ

文字の間隔を操作することによりページ記述に電子透かしを埋め込む手法を提案した. 本手法では, 物理的な文字オブジェクトの情報をレイアウト構造に構成するとともに埋め込みプリミティブである文字オブジェクトを順序付けし, オリジナル文書無しでの検出や統計的手法の導入を可能にしている.

限定されたインプリメントではあるが, PDF文書への埋め込み/検出が可能であることを確認した. ワープロやフォーマッタが個別の文字の位置を操作している文書へも埋め込みが可能であるとの知見を得ることができた. 多様な文書への適用可能性を検証することが今後の課題である.

意図的な攻撃に対して本手法をどこまで頑健にできるかは, より詳細に検討していく必要がある. また, 埋め込み/検出のパフォーマンスについても今後, 評価・改良を行っていききたい.

参考文献

- [1] 沼尾雅之 清水周一 森本典繁: データハイディングによるデジタル署名技術, 情報第53回全国大会 1M-13, (1996).
- [2] Bienz, T. Cohn, R. and Meehan, James R.: Portable Document Format Reference Manual Version 1.2, Adobe Systems Inc. (1996).
- [3] 渋谷竜二郎 梶勇一 嵩忠雄: PostScript および PDF 文書に対するデジタル透かしの提案, SCIS'98 9.2.E, (1998).
- [4] 同定コード埋込み装置, 特開平 6-324625.
- [5] ドキュメントコピー防止方法, 特開平 7-222000.
- [6] 大淵竜太郎 増田宏 青野雅樹: 3次元データへの情報の埋め込み, 情報処理学会 グラフィクスとCAD 研究会/画像電子学会 Visual Computing '97 合同シンポジウム, (1997)
- [7] 平山唯樹: 複雑なカラム構造をもつ文書イメージの領域分割法, 信学論 (D-II), Vol. J79-D-II, No. 11, pp. 1790-1799, (1996)
- [8] 小出昭夫: Data Hiding 技術とその応用, 精密工学会 画像応用技術専門委員会研究報告 Vol. 12 No. 4, pp. 26-33, (1998).
- [9] 吉田淳 岩村恵市 今井秀樹: 画質劣化が少なく結託攻撃に強い電子透かし法, SCIS'98 10.2.A, (1998).